American University of Armenia
College of Science and Engineering
Gor Mkrtchyan
Inna Krmoyan
Irena Torosyan
Fall, 2022

Table of contents

Table of contents 2

Abstract 3

Problem Setting and Description 4

- Fictional Problem Setting 4
- The technical part of the problem formulation 6
- Peas Description 8
- World Characterization 9

Literature review 10

- A SNePS Approach to The Wumpus World Agent or Cassie Meets the Wumpus [1] 10
- Hunt the Wumpus: an Empirical Approach [2] 12
- Agents that reason logically / Wumpus World [3] 14

Implemented Algorithms 15

- The First Algorithm: Brute Force 15
- The Second Algorithm: Finding the PSS location and only attending it 17
- The Third Algorithm: Finding the PSS location and attending both PSS and Study room 22

References 27

Abstract

This paper discusses the definition and the solution of the newly defined game "*Passing AI: Achieving the Impossible"*. Different approaches for solving similar problems are discussed later in the paper as well as their applications in this particular problem. The approaches are modified and tuned and combined for the sake of finding a suitable solution.

Keywords: Artificial Intelligence, Search Problems, Logic, Wumpus World

Problem Setting and Description

Fictional Problem Setting

A regular AUA student has registered for the course “CS246: Artificial Intelligence”. They did not expect the obstacles that were yet to come along their way to the final destination which was passing the course.

After registering for the course the student unexpectedly spawned on a board. The cell that they were in had their bed. They figured that it was the only place they could rest a little. After getting adapted to the environment, the student looked around them and noticed an unusual Cat sitting in the corner of the room. They approached the Cat and it suddenly started talking

- Meow, you finally came to your senses. Other students took way less time than you to figurrre things out. Meow, pathetic mrm. But it is my mrrresponsibility to help you in any case. I’ll keep it short, yourrr study mrrroom is located at (2, 3).

The location of the study room was taken for the sake of the example, the location varies from student to student.

The student had no choice but to trust the arrogant Cat. Afterwards, the student decided to open their laptop as they understood that the Cat had no intention of helping them anymore. They noticed that the course offers Problem Solving Sessions (PSS). Unfortunately, the location was not mentioned.

- Meow, you arrre way more pitiful than I thought. Mrrrm fiiine, I will help you again. Yourrr TA is obsessed with coffee. You will smell the strrrong arrroma of coffee as you get closerrr to the PSS location. Meow.

There is only one PSS location on the board and it never changes. The adjacent cells of the PSS cell and the PSS cell itself are the ones that have the strongest smell of coffee. The smell fades away as the player gets further. The PSS location may vary for each player.

After reading the syllabus of the course and being lectured by the Cat the student realized that they have to stay focused by all means and not be lured by the parties. The student knew how to identify parties as they used to have a peaceful life before the registration. They knew what a good party smells like. It’s pure vodka!

There is only one party location on the board and it never changes. The adjacent cells of the party cell and the party cell itself are the ones that have the strongest smell of vodka. The smell fades away as the player gets further. The party location may vary for each player. Attending a party will lead to an instant and horrible death (i.e. failing the course).

With the newly gained knowledge, the student proceeded to the terrifying semester. They struggled for 4 tedious months and finally, the Cat spoke for the last time.

- Mrrm, I guess you arrre not as bad as I thought you werrre. It is time for me to tell you about the final Boss… Monika, the First of Her Name, Queen of the Logic, Protector of the Vacuum Cleaners, the Mother of Agents, the Unbeatable, the Breaker of Constraints. Will you be able to defend your honorrr?

At the end of the semester, the student passes the course if they have gained enough points. Otherwise, their fate becomes a horrible death.

The technical part of the problem formulation

The view of the board and the location indices.

(0, 4)	(1, 4)	(2, 4)	(3, 4)	(4, 4)
(0, 3)	(1, 3)	(2, 3)	(3, 3)	(4, 3)
(0, 2)	(1, 2)	(2, 2)	(3, 2)	(4, 2)
(0, 1)	(1, 1)	(2, 1)	(3, 1)	(4, 1)
(0, 0)	(1, 0)	(2, 0)	(3, 0)	(4, 0)

Fig. 1: The board with the indices.

Locations:

The bed is located in the cell (0, 0). The location is always known to the player.

The location of the study room is fixed for every student. It may differ from student to student. The location is (S_x, S_y). The location is always known to the player.

The location of the PSS is fixed for every student. The location is unknown to the student. Only one PSS exists at a given time. If the player is in an adjacent cell to the PSS or in the cell of the PSS itself, they will perceive the strongest smell of coffee. The smell will fade away as the player gets further. In this particular definition, there are two degrees of the coffee smell: strong coffee smell and weak coffee smell.

The location of the party is fixed. Only one party exists on the board. If the player is in an adjacent cell to the party or in the cell of the party itself, they will perceive the strongest smell of vodka. The smell will fade away as the player gets further. In this particular definition, there are two degrees of the vodka smell: strong vodka smell and weak vodka smell.

When in a cell with the strongest smell, the student will be able to overcome the lure of the smell only when the proportion {current grade / current time} $> \frac{1}{8}$ holds. When the student is in a cell with a weak smell of vodka, they will overcome the lure without any problem.

After the player has exhausted their moves, they have to face the final boss Monika. The location of the student when they are out of moves does not matter. If the final score of the student is at least 50 then they win the game, otherwise, they suffer a horrible death.

Actions:

Per each 1 point of time spent on sleeping the player gains 2 points of stamina. The action of sleeping does not consume any stamina. The initial points of stamina of the player are 0.

Each move costs the player 1 point of stamina and takes 1 point of time. The lowest stamina that the player can get is 0. The only action available for the player with 0 stamina is sleep. If the player can not perform that action or they choose not to then they suffer a bit less of a horrible death, but still.

Each time point of study consumes 3 points of stamina. The student can study as long as their stamina allows them to do so.

Each PSS lasts for 4 points of time and consumes 4 points of stamina overall.

Score (Grade):

The student gains score points by two means: studying, and attending PSS sessions.

For 1 time point of studying the student gains 4 score points.

For 1 time point of the PSS session, the student gains 2 score points.

The student is allowed to gain more than 100 overall score points if they manage to do so by using some unknown black sorcery.

In the actual game process the player's score is multiplied by 5 at the end of the game, meaning the maximum desirable score is 20.

Time:

The game lasts for 50 time points.

Peas Description

Perfomrance measure (stamina): walk -1, study -3, sleep +2, PSS -4

Performance measure (time): walk +1, study +1, sleep +1, PSS +4

Performance measure (grade): study +1, PSS +8

Environment: Squares adjacent to Party have a strong vodka smell, squares whose horizontal or vertical distance from the party is equal to 2 have a weak vodka smell, squares adjacent to PSS have a strong coffee smell, squares whose horizontal or vertical distance from the PSS is equal to 2 have a weak coffee smell, attend PSS iff you are in the corresponding square, study iff you are in the corresponding square, death if you are in the same square as Party iff {current grade / current time} $< \frac{1}{8}$, Sleep iff you are in the same square as Bed.

Actuators: The player's limbs

Actions: Walk up, Walk down, Walk right, Walk left, Sleep, Study, Attend PSS

Sensors: The player's eyes, nose

Percepts: Weak vodka smell, Strong vodka smell, Weak coffee smell, Strong coffee smell, current location, party, PSS, study room

World Characterization

Partially observable – Only local perception (varies depending on the implemented agent).

Deterministic – Outcomes are exactly specified.

Static – The study room and PSS locations do not move.

Discrete - The board has a limited size, the applicable actions list is also limited.

Single-agent - The only playable character is the Student.

Known - The game rules are known to the agent.

Literature review

Since the problem that we are going to solve was defined by us, there is no literature regarding the solution of the particular problem. We have come to the conclusion that the game “Wumpus World” has the closest resemblance to the newly defined problem. The game “Wumpus World " is usually solved using logic. Nevertheless, different solutions for the problem were reviewed for the sake of combining some strategies and adding new approaches to solve our problem. We are going to combine local search, constraint satisfaction problem solution methods and logic for finding possible strategies for solving our problem.

A SNePS Approach to The Wumpus World Agent or Cassie Meets the Wumpus [1]

One approach of solving problems that have a similar structure to the Wumpus world is using models that have a general human-level intelligence. Such an approach might be more realistic and appropriate while trying to analyze human behavior or create an agent that will compete with a human compared to agents that maximize the use of computing powers.

CassieW is one of the versions of such agents and she implements grounded layered architecture with integrated reasoning. The agent uses the following layers:

- The knowledge layer (KL) is responsible for the “conscious” reasoning. The KL is implemented in SNePS (more about SNePS in the next paragraphs).
- The Perceptuo-Motor Layer, Sublayer a (PMLa) implements actions that are primitive at the KL. The layer accounts for the design of the agent, but is independent of the agent’s body and the environment.
- The Perceptuo-Motor Layer, Sublayer b (PMLb) implements the functionality of the PMLa while accounting for the environment and the agent’s body.
- The Perceptuo-Motor Layer, Sublayer c (PMLc), The Sensori-Actuator Layer (SAL); and The Environment is a general implementation using Java.

The SNePS knowledge base contains the beliefs of the agent as opposed to the information about the agent. The agent's decisions and actions are based on first-person beliefs. The general logic of its operation is that it does not take into account whether or not something is true in the world, but that the agent has enough “reasons” to believe it.

The on-line acting system is implemented. Sensing, reasoning, and inferring is done on-line, during the process of agent's acting.

The two mental actions are believe and disbelieve. The agent has the option of contemplating the propositions that it does not believe in. The propositions that CassieW believes in are a subject to change.

When some new information is inferred a question of whether or not it should be saved arises. The agent faces a space-time trade off. The derived information that may or may not be saved in the KB is referred to as a **lemma**. SNePS saves lemmas in the KB.

The agent does not possess a model of time and has situation-independent beliefs.

The domain entries recognized by the agent are propositions, acts and policies. The latter connects propositions and the acts.

Results:

The outcomes of the experiment are an important measure. They are used for understanding whether or not the newly suggested approaches are worth implementing or not.

The game layouts are divided into 3 groups

- P - possible to win while playing safe
- I - impossible to win
- PS - possible to win only if the agent utilizes risky moves

The results of conducting 10 experiments per type are shown below:

Type	Nwon	AvgWon	Nlost	AvgLost	Avg
P	9	985.0	1	-2023.0	684.2
PS	2	982.0	8	-2006.5	-1408.8
I	0		10	-2008.5	-2008.5

Notice: the usual Wumpus world rules are implemented

Hunt the Wumpus: an Empirical Approach [2]

Another approach of solving the problems that originated from the Wumpus World is using Empirical Modeling that is making artifacts to support human thinking so that it focuses mainly on observations rather than theory. Methods for exploring the performance of the various inferences given different sets of rules and the validity of strategies given an unexpected event occurs are discussed for a classic game of Wumpus. The only difference is that in this definition there are also bats. If the player is in a cell with bats, they can take the player and drop the player at any random location on the board.

There are several types of models that are based on the Evaluator of Definitive Notations (EDEN) which is a tool for Empirical Modeling. The good thing about EDEN is that it is dependency-driven so that certain observables can be defined in terms of others rather than limiting the game to a certain number of states. With the help of EDEN triggered actions can be used to recalculate the probability of a hazard being in a certain room *n,* when some noticeable change occurs. The Empirical modeling tool EDEN provides means of dependencies such as the probability of a Wumpus being in a certain room n depending on whether it was visited before or not or whether there was detected a Wumpus in the adjacent cells of n and other observables. There are texts that appear on the labels that display the made inferences which guide the user throughout the game. Those texts are updated atomically depending on the actions of the user. Here lies the idea of "Definitive programming" which means that the texts displayed are dependent on the experience of the agent and the observables. This implies that the user may perform actions that were not planned by the author.

Strategies, rules, and the outcomes

- The first strategy used is playing the game safely which means exploring the cells that are inferred to contain no danger, but this will not always be a good strategy as the user may spawn in a cell that is adjacent to a pit or other hazard therefore, they have to make a risk. But the experiment proved that it is too easy to win the game given a map and inferences.
- Removing the map makes the game more interesting as the player must make harder inferences and create a map on their own even though the player is supported by the computer which does most of the work by detecting the probabilities of hazards being in nearby cells. The optimal strategy here becomes again exploring as many rooms to avoid hazards as possible.
- Removing both map and inferences will result in a very challenging game as the player must come up with their own inferences about the dangers and the map itself.

- Making inferences fail. Using other adventurous Wumpus game ideas to change the locations of the bats and the Wumpus itself by causing an "earthquake". With the help of EDEN the user can change the locations of the hazards, and all the variables that are dependent on that action will be updated automatically. But on the other hand, the inference table is always dependent on past and current experiences. Therefore this implies that with the user changing the location of a Wumpus, a bat, or a pit, the inference table can give incorrect conclusions.
- Another popular implementation is when the Wumpus moves to another room when the player misses their shot (with an arrow). But this will make all the previous inferences about the Wumpus useless, therefore, a solution to this is making the inference table in a way that when the player misses a shot, the inference table renews the information about the possible locations of the Wumpus.

It, therefore, contributes to the study of how rudimentary AI rules can be constructed using dependencies and triggers in EDEN, and how such EM concepts are useful for game construction. It was shown how the game model represents a player agent interacting with the world and making inferences based on its experiences.

Notice: the usual Wumpus world rules are implemented

Agents that reason logically / Wumpus World [3]

Another approach to solving problems like the Wumpus world is the approach Russel and Norvig (AIMA (4th ed.) [Russell, Norvig 2021]) and some other scientists took, one of which will be discussed.

This approach uses pure logic to find the solution to the problem by using the agent's perceptions and playing safe, without adding any other approach to the problem. The goal of the agent in this case will be getting the gold by using their perceptions and getting out of the maze by going to the cell [1, 1]. The location of the gold and the wumpus are chosen randomly, with a uniform distribution from the squares other than [1, 1]. In addition, each square other than the start location can be a pit, with probability 0.2.

After testing the world multiple times, the author comes to the conclusion that in most environments in this class, it is possible for the agent to return successfully with the gold. However, in some, the agent must choose between going home empty-handed or taking a chance that could lead to death or the gold. And in about 21% of the environments, there is no way the agent can get a positive score (either because they run out of steps or the cell with the gold is spawned with 4 adjacent pits).

The author randomizes the world and in that particular example tries to see whether a solution is present. After doing several hand-on steps, the author realizes that the solution is indeed present and finds the location of both the wumpus and the gold. Afterwards, the author tries using propositional logic in order to find out whether they will come to the same solution as before. In fact, they end up doing the same exact steps and come to the same conclusion. Overall, we can conclude that using both logic and propositional logic are safe and will eventually come to the solution if the agent doesn't have to choose between going home empty handed or taking a risk or if the solution actually exists.

Notice: the usual Wumpus world rules are implemented

Implemented Algorithms

The First Algorithm: Brute Force

The agent ignores every percept and moves towards the study room. Such behavior results in the perception of the world as unobservable and static. This means that the agent can completely plan all the actions and only after that process proceed to their execution. The problem turns into a pure csp. The agent chooses the actions in the following manner:

- Insures that it will not face a horrible death because of exhaustion (not enough stamina)
- Makes sure to plan for the whole period of time and use it as efficiently as its stubborn logic allows it to do.
- By calculating the distance that needs to be traversed, the distance from the study room and the stamina for the mentioned actions, moves to the desired location.
- If the agent bumps into deadly obstacles, such as the luring smell of alcohol, it dies.
- The agent moves to the right as much as needed, afterward continues moving upwards until it reaches the study room.

P.S. We'd like to blame the agent's death on its own stubbornness and not our poor decision of using such a simple algorithm. We do not take any responsibility for the agent's death. Use all the calculations at your own risk.

Here are the results of 20 random simulations of the game.

Predicted Score	Pred. Score/100	Actual Score	Act. Score/100	Win/Loss	Death Cause
19	95	19	95	Win	
16	80	0	0	Loss	Strong Vodka
16	80	0	0	Loss	Strong Vodka
17	85	17	85	Win	
18	90	18	90	Win	
18	90	0	0	Loss	Strong Vodka
17	85	0	0	Loss	Strong Vodka
19	95	19	95	Win	
16	80	0	0	Loss	Strong Vodka
17	85	17	85	Win	
18	90	18	90	Win	
17	85	17	85	Win	

16	80	16	80	Win	
18	90	0	0	Loss	Strong Vodka
17	85	17	85	Win	
18	90	18	90	Win	
15	75	0	0	Loss	Strong Vodka
18	90	18	90	Win	
17	85	17	85	Win	
17	85	0	0	Loss	Strong Vodka
17.2	86	10.55	52.75	12/8	

Conclusions:

- The agent always succeeded in providing a path that should be followed.
- The provided path was a solution in 60% of the cases
- By assigning 0-s to the lost games we obtained an average score or 52.75 which is just over the passing grade.
- While the algorithm's performance is extremely far from ideal, considering its simplicity, it performed better than we expected.
- The performance of the algorithm is a good base for choosing whether or not the more complicated algorithms are worth implementing.

The Second Algorithm: Finding the PSS location and only attending it

The agent uses a search algorithm for moving towards the PSS, while checking that the constraints are satisfied and choosing actions based on some predefined logic.

The agent goes through the steps after each action and chooses the next action and returns it. (1 action at a time). The decisions are made on-line, during the process of acting.

From this point on let's call the agent Murad. (The name was chosen randomly.)

The reasoning of Murad

- Murad ignores percepts related to the party and hopes that he will not die.
- Murad keeps track of not dying because of the lack of stamina.
- Murad does not visit the study room and only tries to visit the PSS.

Keeping track of stamina

- When Murad is in the location of the bed he sleeps until he has at least 20 points of stamina.
- When Murad senses the luring strong smell of the alcohol and he does not have a high enough grade he accepts his horrible death.
- When Murad is choosing what action to perform (attend pss, walk, etc.) he makes sure that after performing the said action he will be able to walk back to his bed and not die because of the lack of stamina.
- Murad completely ignores the percepts related to the party.

Choosing a square to move to

- Murad takes care of not getting out of the board
- Case 1: Murad senses a weak coffee smell and he has never sensed any pss related smells before
    - Murad moves to the adjacent cells moving clockwise, starting from the right cell.
    - If Murad senses a strong coffee smell he moves to Case 2.
    - If Murad senses nothing he comes back to the previous cell and continues exploring the adjacent cells.
- Case 2: Murad senses a strong smell of coffee
    - Murad moves to the adjacent cells moving clockwise, starting from the right cell.
    - During the search of the PSS room, Murad does not move to the previously explored rooms, since he knows that the PSS can not be contained there.

- If Murad senses a PSS, he attends it (if he has enough stamina).
- If Murad does not sense a PSS, he comes back to the previous cell and continues exploring the adjacent cells.

Returning back

- Murad returns back by the shortest, already exploded path (in the manhattan distance sense), thinking that it is the safest.

Getting back to the pss

- Murad returns to the PSS by the shortest, already exploded path (in the manhattan distance sense), thinking that it is the safest.

Note: While the score may get over 100 it is not going to be considered anymore. So, let's say, if Murad has already gained a score of 100 and he gains additional 20 points the score remains 100.

Note: Since the score is multiplied by 5 at the end of the game, in Murad's calculations, the highest achievable grade should be set to 20.

Example:

Here's an example of a randomly generated world.

{(0, 0): ['bed', 'current location'],
(0, 1): ['weak coffee smell'],
(0, 2): ['weak vodka smell'],
(0, 3): ['strong vodka smell'],
(0, 4): ['party', 'strong vodka smell'],
(1, 0): [],
(1, 1): ['study room', 'strong coffee smell'],
(1, 2): [],
(1, 3): [],
(1, 4): ['strong vodka smell'],
(2, 0): ['strong coffee smell'],
(2, 1): ['pss', 'strong coffee smell'],
(2, 2): ['strong coffee smell'],
(2, 3): ['weak coffee smell'],
(2, 4): ['weak vodka smell'],
(3, 0): [],
(3, 1): ['strong coffee smell'],

(3, 2): [],
(3, 3): [],
(3, 4): [],
(4, 0): [],
(4, 1): ['weak coffee smell'],
(4, 2): [],
(4, 3): [],
(4, 4): []}

In this particular case, Murad's actions and some statistics throughout the game are:

Action	Time	Stamina	Grade
Sleep x10	10	20	0
Walk to (1, 0)	11	19	0
Walk to (1, 1)	12	18	0
Walk to (2, 1)	13	17	0
PSS	17	13	8
PSS	21	9	16
PSS	25	5	20
Walk to (1, 1)	26	4	20
Walk to (1, 0)	27	3	20
Walk to (0, 0)	28	2	20
Sleep x9	37	20	20
Walk to (1, 0)	38	19	20
Walk to (1, 1)	39	18	20
Walk to (2, 1)	40	17	20
PSS	44	13	20
PSS	48	9	20

Murad has survived with an unbelievable grade of $20 * 5 = 100$.

The results:

Predicted Score	Pred. Score/100	Last Best Sense	Actual Score	Act. Score/100	Win/Loss	Death Cause
0	0	None	0	0	Loss	Strong Vodka
0	0	Strong Coffee	0	0	Loss	Strong Vodka
20	100	PSS	20	100	Win	
0	0	None	0	0	Loss	Strong Vodka
20	100	PSS	20	100	Win	
20	100	PSS	20	100	Win	
0	0	PSS	0	0	Loss	Strong Vodka
0	0	None	0	0	Loss	No Solution
20	100	PSS	20	100	Win	
0	0	PSS	0	0	Loss	Strong Vodka
8	40		8	40	4/ 10	

- We can see that Murad has predicted his score to be 0 in the cases when he lost. The reason is online acting. This can be viewed as an advantage, since he does not continue computations when after performing some action he realizes that something has caused a horrible death. In this case the predicted score is 0 because he realizes that he is going to die. (That's quite depressing, but relatable.)
- Since Murad explores more cells and travels more, he has more chances to die because of the luring alcohol smell (as opposed to the first algorithm).
- The effectiveness of the algorithm is highly dependent on the pss location, which is random. This makes the algorithm not very reliable.

Explanation of the 7th trial and 10th:

- Murad managed to find the PSS location but it had a strong vodka smell… :')

Explanation of the 8th trial:

- It is worth mentioning that the algorithm does not guarantee finding the location of the PSS. In this trial, for example, Murad failed to perceive any information about the location of the room.
- The reason why we think it is not an issue for this algorithm is because the only cases when Murad will not get any percept is when the pss is on some specific edges on a far distance. We think that if we made the search algorithm any more complicated, we would have a much higher risk of running out of time or dying because of the vodka smell. The latter is resulted by traversing a larger amount of cells.

Conclusions:

- Even though in the cases when Murad succeeded, he has achieved the highest possible score, he has only succeeded in 40% of the cases.
- Given how much more computational potential this algorithm requires we think that is barely a sidegrade as compared to the first, simplest algorithm.
- However, the useful information that we obtained while designing and using this algorithm is:
    - The problem has quite a few unsolvable configurations.
    - In this particular definition of the game, the search is more punishing than rewarding (complex algorithms, higher risk of traveling vs higher grade obtained from attending the PSS).

The Third Algorithm: Finding the PSS location and attending both PSS and Study room

The agent starts moving towards the Study room, if on the way to the Study room, it senses any kind of coffee smell, it starts using a search algorithm for moving towards the PSS, while checking that the constraints are satisfied and choosing actions based on some predefined logic.

The agent goes through the steps after each action and chooses the next action and returns it. (1 action at a time). The decisions are made on-line, during the process of acting.

From this point on for this particular case we'll call our agent Malvina. (The name was chosen randomly)

The reasoning of the agent

- Malvina ignores percepts related to the party and hopes that she will not die.
- Malvina keeps track of not dying because of the lack of stamina.
- Malvina is able to visit both PSS (in case she finds it, as the location is unknown) and the Study room
- Malvina will try to search for the PSS on the way to the Study room. If on the way to the Study room she doesn't sense any kind of coffee smell, she will give up and go to the Study room.

Keeping track of stamina

- When Malvina is in the location of the bed she sleeps until she has at least 20 points of stamina.
- When Malvina senses the luring strong smell of the alcohol and she does not have a high enough grade she accepts his horrible death.
- When Malvina is choosing what action to perform (attend PSS, walk, study, etc.), she makes sure that after performing the said action she will be able to walk back to her bed and not die because of the lack of stamina.
- Malvina completely ignores the percepts related to the party.

Choosing a square to move to

- Malvina takes care of not getting out of the board
- Case 1: On the way to the Study room, Malvina senses a weak coffee smell and she has never sensed any PSS related smells before
    - Malvina moves to the adjacent cells moving clockwise, starts from the right cell
    - If Malvina senses a strong coffee smell he moves to the Case 2

    - If Malvina senses nothing he comes back to the previous cells and continues exploring the adjacent cells
- Case 2: Malvina senses a strong smell of coffee
    - Malvina moves to the adjacent cells moving clockwise, starts from the right cell
    - During the search of the PSS room, Malvina does not move to the previously explored rooms, since she knows that the PSS can not be contained there.
    - If Malvina senses a PSS she attends it (if she has enough stamina)
    - If Malvina does not sense a PSS, she comes back to the previous cells and continues exploring the adjacent cells
- Case 3: Malvina doesn't sense any Coffee Smell on the way to the Study room
    - Malvina continues her way to the Study room, and proceeds studying there until she has just enough stamina to get back to the bed or until the end of the game (if she has enough stamina)

Returning back

- Malvina returns back to the bed by the shortest, already exploded path (in the manhattan distance sense), thinking that it is the safest

Note: While the score may get over 100, it is not going to be considered anymore. So, let's say, if Malvina has already gained a score of 100 and she gains additional 20 points, the score remains 100.

Note: Since the score is multiplied by 5 at the end of the game, in Malvina's calculations, the highest achievable grade should be set to 20.

Example:

Here's an example of a randomly generated world.

{(0, 0): ['bed', 'current location'],
(0, 1): ['strong coffee smell'],
(0, 2): [],
(0, 3): ['study room', 'weak vodka smell'],
(0, 4): [],
(1, 0): ['strong coffee smell'],
(1, 1): ['pss', 'strong coffee smell'],

(1, 2): ['strong coffee smell'],

(1, 3): ['strong vodka smell', 'weak coffee smell'],

(1, 4): [],

(2, 0): [],

(2, 1): ['weak vodka smell', 'strong coffee smell'],

(2, 2): ['strong vodka smell'],

(2, 3): ['party', 'strong vodka smell'],

(2, 4): ['strong vodka smell'],

(3, 0): [],

(3, 1): ['weak coffee smell'],

(3, 2): [],

(3, 3): ['strong vodka smell'],

(3, 4): [],

(4, 0): [],

(4, 1): [],

(4, 2): [],

(4, 3): ['weak vodka smell'],

(4, 4): []}

In this particular case, Malvina's actions and some statistics throughout the game are:

Action	Time	Stamina	Grade
Sleep x10	10	20	0
Walk to (0, 1)	11	19	0
Walk to (1, 1)	12	18	0
PSS	16	14	8
PSS	20	10	16
PSS	24	6	20
PSS	28	2	20
Walk to (0, 1)	29	1	20
Walk to (0, 0)	30	0	20

Sleep x10	40	20	20
Walk to (0, 1)	41	19	20
Walk to (1, 1)	42	18	20
PSS	46	14	20
PSS	50	10	20

Malvina has passed with an outstanding grade of 100!

The results:

Predicted Score	Pred. Score/100	Last Best Sense	Actual Score	Act. Score/100	Win/Loss	Death Cause
20	100	PSS	20	100	Win	
0	0	None	0	0	Loss	Strong Vodka
20	100	None	20	100	Win	
0	0	PSS	0	0	Loss	Strong Vodka
20	100	PSS	20	100	Win	
20	100	PSS	20	100	Win	
20	100	None	20	100	Win	
0	0	Strong Coffee	0	0	Loss	Strong Vodka
0	0	None	0	0	Loss	No solution
20	100	None	20	100	Win	
12	60		12	60	6/10	

- We can see that Malvina has predicted her score to be 0 in the cases when she lost. The reason is online acting. This can be viewed as an advantage, since she does not continue computations when after performing some action she realizes that something has caused a horrible death. In this case the predicted score is 0, because she realizes that she is going to die. (That's quite depressing, but relatable.)
- The main reasons for Malvina's failure are because on the way to the Study room, she either sensed Strong Vodka smell or she sensed any kind of coffee smell, and started searching for the PSS. Because of the search, she sometimes came across Strong Vodka smell and died.

- The effectiveness of the algorithm is highly dependent on the Study room and PSS locations, which is random. This makes the algorithm not very reliable.
- It's still more reliable than the second algorithm, even though it's not the best.

Explanation of the 4th trial:

- Malvina managed to find the PSS location, but it had a strong vodka smell.

Explanation of the 3rd, 7th and 10th trial:

- Malvina didn't search for the PSS, as she didn't come across any type of coffee smell on the way to the Study room. She simply went to the Study room, studied as much as the stamina would allow her to, then she went to the starting cell taking the same route. She repeated these actions until the end of the time limit.

Conclusions:

- Even though in the cases when Malvina succeeded, she has achieved the highest possible score, she has only succeeded in 60% of the cases.
- This algorithm is a positive improvement compared to the first two algorithms.
- Even though this algorithm also implements search, the agent spends less time on the search and explores less cells (as opposed to the second algorithm), which results in less cases when she died because of the party.
- The algorithm has a good balance between searching for the PSS and going to her desired location.

References

[1] Shapiro, S. C., & Kandefer, M. (2000). A SNePS Approach to The Wumpus World Agent or Cassie Meets the Wumpus. cse.buffalo.edu. Retrieved November 14, 2022, from https://cse.buffalo.edu/~shapiro/Papers/shakan05a.pdf

[2] Hunt the Wumpus: an Empirical Approach. (n.d.). warwick.ac.uk. Retrieved November 14, 2022, from https://warwick.ac.uk/fac/sci/dcs/research/em/publications/web-em/01/wumpus.pdf

[3] Agents that reason logically. (n.d.). Massey.ac.nz. Retrieved November 14, 2022, from https://www.massey.ac.nz/%7Emjjohnso/notes/59302/l06.html